

# **APLIKASI TEORI GRAF DAN ALGORITMA BACKTRACK DALAM PENYELESAIAN MASALAH PERJALANAN BIDAK KUDA (KNIGHT'S TOUR) PADA PAPAN CATUR**

Oleh:  
Srianto  
NIM. 05305144050

## **ABSTRAK**

*Knight's tour* pada papan catur adalah suatu rangkaian perjalanan bidak kuda catur pada papan catur sehingga seluruh kotak terlewati oleh kuda tepat satu kali. Menurut jenisnya *Knight's tour* dibedakan menjadi dua yaitu *Open Knight's tour* dan *Close Knight's tour*. *Open Knight's tour* adalah perjalanan bidak kuda yang harus mampu melewati semua kotak pada bidang catur tepat satu kali dan tidak kembali ke kotak awal. Dalam permasalahan teori graf hal itu dikenal dengan *lintasan Hamilton*, sedangkan *Close Knight's tour* adalah perjalanan bidak kuda yang harus melewati semua kotak bidang tepat satu kali dan harus dapat kembali ke kotak awal. Dalam teori graf hal itu dikenal dengan *siklus Hamilton*.

Banyak algoritma yang telah ditemukan untuk menyelesaikan *Knight's tour*, salah satunya algoritma *Backtrack*. Algoritma *Backtrack* merupakan suatu algoritma perbaikan dari algoritma Brute Force dengan menggunakan algoritma rekursif dan berbasis pada DFS (Depth-First Search) dalam mencari solusi. Selain itu, algoritma ini juga merupakan metode yang mencoba-coba dari beberapa keputusan sampai menemukan salah satu solusi. Algoritma *Backtrack* ini mampu mencari ada atau tidak adanya solusi *Close Knight's tour* dan *Open Knight's tour*.

Pada pembahasan skripsi ini lebih difokuskan pada proses penyelesaian *Open Knight's tour* dan *Close Knight's tour* pada papan catur berukuran  $m \times n$  dengan  $m \leq 12$  dan  $n \leq 12$  menggunakan algoritma *Backtrack*. Penyelesaian *Open Knight's tour* dan *Close Knight's tour* menggunakan algoritma *Backtrack* menghasilkan solusi berupa *lintasan Hamilton* atau *siklus Hamilton* pada papan catur. Penyelesaian *Knight's tour* ini memunculkan beberapa kasus diantaranya adalah karakter papan catur yang mempunyai atau tidak mempunyai solusi. Selanjutnya adalah kasus jika dimulai dari kotak awal yang berbeda dan solusi tunggal atau tidak tunggal. Jika dapat ditemukan solusi *Close Knight's tour* pada suatu kotak sebagai kotak awalnya maka akan dapat ditemukan juga solusi *Close Knight's tour* di kotak lain sebagai kotak awalnya. Hal itu tidak berlaku pada *Open Knight's tour*, kotak awal dimulainya perjalanan kuda sangat mempengaruhi ada atau tidak adanya solusi *Open Knight's tour* pada papan catur.

Algoritma *Backtrack* dapat dibuat menjadi perangkat lunak, sehingga proses pencarian solusi *Open Knight's tour* dan *Close Knight's tour* akan lebih cepat dibandingkan dengan pencarian secara manual. Hasil yang didapatkan antara pencarian dengan program algoritma *Backtrack* sama dengan pencarian secara manual.